

REMARKS

Reconsideration of this application, based on this amendment and these following remarks, is respectfully requested.

Claims 1 through 6, 9, 11, 20 through 30, and 33 remain in this case. Claims 1, 23 through 30, and 33 are amended.

Claim 1 and its dependent claims 2 through 6, 8, 9, 11, and 20 through 22, were again rejected under §112, ¶2, as indefinite for failing to particularly point out and distinctly claim the subject matter of the invention. Specifically, the Examiner asserted that claim 1 was still unclear because the determining step appears to include a plurality of data processors in the system, but that the downloading step refers to "the data processor that satisfies said condition".¹ Dependent claims 2 through 6, 8, 9, 11, and 20 through 22 were rejected as depending on claim 1, so rejected.

Claim 1 is amended to overcome this rejection, by now reciting that the system has at least one data processor, that the determining step determines whether at least one data processor in the system satisfies the at least one platform requirement, and that the downloading step downloads the program to one of the at least one data processors that satisfies the platform requirement. Support for this recitation is clearly provided by the specification,² and therefore no new matter is presented by this amendment. Applicant submits that amended claim 1 now expressly and repeatedly states that the system may have one or more data processors, in a manner that consistently applies to the determining and downloading steps. Furthermore, Applicant therefore respectfully submits that amended claim 1 remains generic to both a single data processor system and also to a system having multiple data processors, because the claim can clearly be read on either case without confusion.

¹ Office Action of August 5, 2004, page 3.

² Specification of S.N. 09/822,753, page 5, line 9 through page 6, line 6.

Applicant respectfully submits that amended claim 1 and its dependent claims are sufficiently definite as to meet the requirements of §112, ¶2.

Claims 25 through 30 and 33 were rejected under §112, ¶2 as indefinite for failing to particularly point out and distinctly claim the subject matter of the invention. Specifically, the Examiner rejected these claims because the preamble in each referred to the "system" of the claim upon which it depends, but that independent claim 24 recites an "apparatus". Each of claims 25 through 30 and 33 are amended to now recite "apparatus" in the preamble, as suggested by the Examiner. Applicant respectfully submits that amended claims 25 through 30 and 33 now meet the requirements of §112, ¶2.

Claims 1 through 4, 6, 9, 11, 21 through 24, 28, 29, and 33 were rejected under §103 as unpatentable over the Halpern et al. reference³ in view of the Bretscher reference⁴. The Examiner asserted that the Halpern et al. reference teaches a method of downloading a program to a data processor, in which the program is in an executable file along with information that inherently indicates "a condition".⁵ The Examiner admitted that the Halpern et al. reference does not explicitly teach the step of downloading the program responsive to a determining step, but asserted that the Bretscher reference discloses a method in which it determined, based on platform requirements, whether to download a program to a data processor.⁶ The Examiner then concluded that one skilled in the art would be motivated to combine these teachings in order to download a program to a data processor in a system having "not homogenous" processors, as suggested by the Bretscher reference.⁷

Claim 5 was finally rejected under §103 as unpatentable over the Halpern et al. and Bretscher references, further in view of the Carron et al. reference.⁸ Claims 25 through 27 and 30

³ U.S. Patent No. 6,282,711 B2, issued August 28, 2001 to Halpern et al., from an application filed August 10, 1999.

⁴ U.S. Patent Application Publication No. US 2001/0039569 A1, published November 8, 2001, on an application filed April 6, 1998 by Bretscher.

⁵ Office Action, *supra*, ¶ 10, page 4.

⁶ *Id.*, at page 4.

⁷ *Id.*, at pages 4 and 5.

⁸ U.S. Patent No. 4,724,521, issued February 9, 1988, to Carron et al.

were finally rejected under §103 as unpatentable over the Halpern et al. and Bretscher references, further in view of the admitted prior art. Claim 20 was finally rejected under §103 as unpatentable over the Halpern et al. and Bretscher references, further in view of the Tevanian et al. reference.⁹

Claim 1 is amended to advance the prosecution of this application, by further clarifying its patentability over the applied references.

Amended claim 1 now requires determining whether a data processor in the system satisfies the at least one platform requirement for the retrieved program, from the non-program information in an executable file in which the program and non-program information is provided. Support for this amendment to claim 1 is present in the specification of this application¹⁰, and therefore no new matter is presented by this amendment.

As previously argued, the method of amended claim 1 provides important advantages in the management of co-processors in multi-processor systems.¹¹ Because the condition information is provided within an executable file containing the program itself according to the claimed method, auxiliary data sources (e.g., SQL engine, Microsoft registry, auxiliary text files) are not necessary and need not be handled to initiate program execution.¹² Furthermore, the claimed method enables a host processor to ensure, before program download, that the destination co-processor is capable of executing the program,¹³ and to select an optimum one of multiple co-processors in the system, if more than one is available¹⁴. Additionally, the method of amended claim 1 avoids the need for separate APIs for multiple operating systems¹⁵, and is sufficiently efficient that it can be used in single-chip systems¹⁶.

⁹ U.S. Patent No. 5,604,905, issued February 18, 1997, to Tevanian et al.

¹⁰ Specification, *supra*, page 5, line 18 through page 6, line 6; page 8, lines 14 through 17; page 9, lines 17 through 21.

¹¹ Specification, *supra*, page 11, lines 6 through 19.

¹² Specification, *supra*, page 3, lines 3 through 8.

¹³ Specification, *supra*, page 9, lines 17 through 21.

¹⁴ Specification, *supra*, page 6, lines 3 through 6.

¹⁵ Specification, *supra*, Figure 8.

¹⁶ Specification, *supra*, page 1, line 15 through page 2, line 2.

Applicant respectfully submits that amended claim 1 and its dependent claims are patentably distinct over the Halpern et al. and Bretscher references. Specifically, Applicant submits that the combined teachings of these references fall short of the claims, because neither reference teaches the claimed step of determining of whether a data processor in a system satisfies platform requirements, from non-program information that is provided in an executable file along with the program itself, much less then downloading the program to the data processor responsive to this determining step, as required by amended claim 1.

As previously argued, the Halpern et al. reference discloses the custom building of a software installation package based on user inputs from a client system, and then sending the customized installation package to the client system.¹⁷ According to the Halpern et al. reference, an executable file is generated *after* information regarding the eventual data processor is gathered and analyzed. The Halpern et al. reference therefore necessarily fails to disclose the determining, from non-program information within the executable file itself, of whether a data processor satisfies at least one platform requirement for a retrieved program, as required by amended claim 1.

The Bretscher reference also lacks teachings in this regard. The cited location of the reference¹⁸ describes the operation of its system in response to a user request to execute a real-time application, beginning with a front-end server sending a message to all dedicated processors (presumably corresponding to the data processors of claim 1) to ask for status information, specifically their availability to accept a new application. The front-end server "then chooses an available dedicated processor 116 that is of the appropriate type and capacity to run the user's selected application" and, if a "disk-farm" is associated with the server, it "retrieves the selected application stored in the disk farm 114", and downloads the application to the chosen dedicated processor after sending messages to the processor to accept the download and configuration information.¹⁹ According to another embodiment of the reference, the "front-end server" does not ever receive the downloaded executable file; rather, the

¹⁷ Halpern et al., *supra*, column 4, line 44 through column 6, line 19.

¹⁸ Bretscher, *supra*, paragraph [0055].

¹⁹ *Id.*

dedicated processors are associated with their own "disk farms", from which they receive the executable files directly, in response to a command from the front-end server.²⁰

Nowhere does the Bretscher reference anywhere disclose that it uses non-program information in an executable file to determine whether any of its dedicated processors "is of the appropriate type and capacity" to execute the application. A fair reading of the Bretscher reference instead leads to the conclusion that its "front-end server" is not in possession of the executable file as it chooses one of the processors (the executable file not yet retrieved from the disk-farm at that point), and therefore is not capable of using non-program information from the executable file to determine whether a processor satisfies a platform requirement, as required by claim 1. Indeed, according to the Bretscher reference, the selection by the front-end server appears to be made from the status information returned from the processors in response to a message asking for that information,²¹ and apparently from some unspecified (and presumably predetermined) source of information regarding "type and capacity". In any case, the Bretscher reference nowhere discloses determining, from non-program information in an executable file that also provides the program to be downloaded, whether a data processor in the system satisfies the platform requirements specified in that information, as required by amended claim 1.

The other references of record in this case also lack teachings in this regard.

Accordingly, Applicant respectfully submits that the combined teachings of the references fall short of the requirements of amended claim 1.

Applicant further respectfully submits that there is no suggestion from the prior art or otherwise to modify these teachings in such a manner as to reach amended claim 1 and its dependent claims. As previously argued, this lack of suggestion is especially apparent when one considers the purpose and intent of the Halpern et al. teachings, which are directed to the generation of a custom installation package for a client system, based on inputs from the user,

²⁰ Bretscher, *supra*, paragraph [0057].

²¹ *Id.*, see also Bretscher, *supra*, claim 4.

so that unnecessary software components are not downloaded to the user's system. This purpose of the Halpern et al. system is actually opposite to that of the claimed method, in which a program is downloaded to a data processor only after determining that the data processor satisfies platform requirements of the program. The distinction between the two is especially evident considering the context of the word "required" in the location of the reference cited by the Examiner, which states that the single download "is no bigger or smaller than what is absolutely required by the components and options selected"²². The word "required", in the Halpern et al. reference, refers to paring down the installation package to only those program components that are *required by the receiving processor*; in contrast, the claimed determining step determines whether the receiving processor meets a platform requirement *required by the program* prior to downloading the program.

The Bretscher reference provides no suggestion to modify its teachings, individually or in combination with the Halpern et al. and other references, so as to reach amended claim 1. In fact, the system disclosed in the Bretscher reference appears to suffer from the very problem addressed by the invention of claim 1, namely that non-program information is stored separately from the executable file containing the program, requiring the use and handling of an auxiliary data source and multiple files.²³ Suggestion to modify the prior art teachings as necessary to reach the claims is therefore lacking from the Bretscher reference, as it is from the other references of record in this case. And because this benefit of the invention of amended claim 1 flows directly from the difference between the claim and the prior art, particularly this Bretscher reference, Applicant submits that this important advantage, as well as the others mentioned above, support the patentability of the claims.

Applicant therefore respectfully submits that amended claim 1 and its dependent claims is be patentably distinct over the applied references.

Independent claim 23 is amended similarly as claim 1, discussed above. Amended claim 23 now recites that the file storage facility includes an executable file containing a program and

²² Halpern et al., *supra*, Abstract, lines 19 through 21, cited in Office Action, *supra*, ¶7.

²³ Specification, *supra*, page 3, lines 3 through 8.

also non-program information comprised of at least one platform requirement for execution of the program. The claimed apparatus now also requires a first data processor that is programmed to perform a sequence of operations, comprising obtaining the program and non-program information from the file storage facility; determining, from the non-program information in the executable file, whether a second data processor satisfies the at least one platform requirement; and responsive to determining that this second data processor satisfies the requirement, then downloading the program to the second data processor.

Independent claim 24 is similarly amended as is claim 23, and is directed to an apparatus having first and second data processors, a file storage facility including an executable file containing a program and non-program information comprising at least one platform requirement for execution of the program. The claimed apparatus also requires that its first data processor is programmed to obtain the program and non-program information from the file storage facility; to determine, from the non-program information in the executable file, whether a second data processor satisfies the at least one platform requirement; and responsive to determining that this second data processor satisfies the requirement, then download the program to the second data processor.

Applicant respectfully submits that amended claims 23, 24, and claims 25 through 30 and 33 that depend on amended claim 24, are patentably distinct over the applied references, on similar grounds as discussed above relative to amended claim 1.

As discussed above, Applicant respectfully submits that the Halpern et al. reference fails to disclose the determining, from non-program information in an executable file that also contains a program, of whether a second data processor satisfies at least one platform requirement for a retrieved program. The Halpern et al. reference instead teaches, at most, the construction of a downloadable software package (*i.e.*, an executable file) in response to user inputs, thus generating an executable file only *after* information regarding the eventual data processor is gathered and analyzed.

Also as discussed above, the Bretscher reference also lacks teachings in this regard. The cited location of the reference²⁴ discloses, at most, that execution of an application occurs by a front-end server asking dedicated processors for their availability to accept a new application, following which the front-end server "chooses an available dedicated processor 116 that is of the appropriate type and capacity to run the user's selected application". But this selection is followed by retrieving the selected application from a "disk farm", and downloading the application to the chosen dedicated processor.²⁵ Elsewhere, the reference discloses another example in which the "front-end server" in fact does not receive the downloaded executable file, but instead commands an associated disk farm to directly forward the executable file to its dedicated processor.²⁶ Therefore, the Bretscher reference fails to disclose using non-program information from the executable file containing the program to determine whether any of its dedicated processors "is of the appropriate type and capacity" to execute the program, and therefore fails to disclose the determining step of claims 23 through 30 and 33.

The other references of record in this case also lack teachings in this regard.

Accordingly, Applicant respectfully submits that the combined teachings of the references fall short of the requirements of amended claims 23 and 24, and dependent claims 25 through 30 and 33.

Applicant further submits that there is no suggestion from the prior art to modify the teachings of the applied references so as to reach these claims. As discussed above, the purpose of the Halpern et al. teachings is in fact opposite from that of the claimed invention, in that Halpern teaches generating the executable file once the data processor is known, rather than determining the data processor from non-program information in the executable file, as claimed. And also as mentioned above, the system of the Bretscher reference suffers from the very problem that is addressed by the invention, namely the handling of multiple files in order to control the downloading and execution of a program to the correct data processor. The

²⁴ Bretscher, *supra*, paragraph [0055].

²⁵ *Id.*

²⁶ Bretscher, *supra*, paragraph [0057].

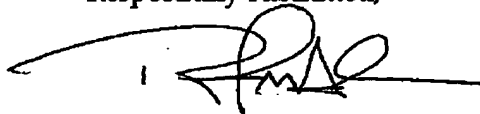
important advantages provided by the apparatus of claims 23, 24, and the claims dependent on claim 24, further support the patentability of these claims, especially considering that these advantages result directly from the difference between the claims and the prior art. Accordingly, Applicant submits that there is no suggestion from the prior art to modify the combined teachings of the applied references to reach these apparatus claims.

For these reasons, Applicant respectfully submits that amended claim 23, amended claim 24 and its dependent claims will be patentably distinct over the applied references.

The prior art cited by the Examiner as pertinent, but not applied, has been considered but is not felt to come within the scope of the claims now in this case.

For these reasons, Applicant respectfully submits that, upon entry of this amendment, all claims remaining in this case will be in condition for allowance. Entry of this amendment in, and favorable reconsideration of, this application are therefore respectfully requested.

Respectfully submitted,



Rodney M. Anderson

Registry No. 31,939

Attorney for Applicant

Anderson, Levine & Lintel, L.L.P.

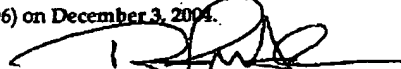
14785 Preston Road, Suite 650

Dallas, Texas 75254

(972) 664-9554

CERTIFICATE OF FACSIMILE TRANSMISSION
37 C.F.R. 1.8

The undersigned hereby certifies that this correspondence is being facsimile transmitted to the Patent and Trademark Office (Fax Number 703.872.9306) on December 3, 2004.



Rodney M. Anderson
Registry No. 31,939